APPLICATION

FOR

UNITED STATES

LETTERS

PATENT

METHOD AND APPARATUS FOR SUMMING ASYNCHRONOUS SIGNAL SAMPLES IN A CABLE TELEVISION DIGITAL RETURN PATH

INVENTORS:

Robert L. Howald
Sean Gallagher
Theodore Booth

ASSIGNEE:

General Instrument Corporation
Invention Record No. D2506

METHOD AND APPARATUS FOR SUMMING ASYNCHRONOUS

SIGNAL SAMPLES IN A CABLE TELEVISION DIGITAL RETURN PATH

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001]   The present invention is related to U.S. Provisional application 60/209,083 filed June 2, 2000, which is incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

[0002]   The present invention relates generally to methods and apparatuses for digitizing multiple data streams, and more particularly to a method and apparatus for digitizing multiple data streams that are geographically diverse.

[0003]   Advances in analog to digital converters (A/Ds) have made transmission of the digitized analog RF return path in a Hybrid Fiber-Coax (HFC) cable system an attractive alternative to analog transmission because digital transmission relaxes the requirement for expensive linear transmit lasers. Moreover, modern digital signal processing (DSP) techniques that are embodied in reconfigurable digital hardware devices, which are known as field programmable gate arrays (FPGAs), can perform processing tasks that were previously relegated to RF devices. Examples of such functions include, *inter alia,* signal adding, filtering, channelizing and demodulating. While analog signal processing functions have a digital counterpart, the digitization process introduces additional flexibility and

tradeoffs that do not have an analog counterpart. Simple examples would be digital

word length (e.g., a number of bits/sample) and sample rate.

[0004]   Once major issue exists that is peculiar to the digital return path of an

HFC system. In such a system, the RF return path could be digitized in the node

and then transmitted digitally to the Hub and/or Head-end. A problem occurs when

two or more digitized signal streams from different nodes are to be added together,

particularly if the nodes are geographically diverse, and hence are subject to

different maintenance schedules and environmental conditions. Ideally, two signal

streams would be sampled at identical sample rates and thus be synchronized prior

to being summed.

[0005]   While each node would have identical sampling clock frequencies

generated from a crystal oscillator, oscillators suitable from a performance and

economic standpoint for HFC may drift up to five parts per million over time and

temperature. For a 100 MHz oscillator that would be used in a 5-40 MHz return

path, this would be equivalent to an oscillator whose actual frequency range could

range from 99.995 MHz to 100.005 MHz. The worst-case difference between two

digital data streams that are to be added would be as much as 10 kHz.

[0006]   Once must then consider how long it would take the synchronizing

first-in-first-out (FIFOs) to underflow or overflow because of the sample rate

difference between the two data streams. Using the numbers above, this can be

shown to be in the 1-2 msec range, depending upon the size of the FIFO buffers.

This would result in the loss of return path data approximately every millisecond,

which results in unacceptable performance.

Docket: D2506

[0007]   To keep the FIFOs balanced (i.e., the input data rate equals to the

output data rate out), one could periodically drop a sample from the input of the

FIFO to keep it from overflowing or periodically repeat a sample at the output of

the FIFO to keep it from underflowing. However, unless the original RF signals are

highly over sampled (by orders of magnitude), periodically dropping or adding

samples will introduce an unacceptably high distortion level such that the data will

be excessively degraded. Current 10-bit A/Ds can be clocked up to 105 MHz,

which is sufficient to satisfy the Nyquist sampling theory, but far less than the

orders of magnitude needed when sampling a 5 MHz to 40 MHz return band.

[0008]   The present invention is therefore directed to the problem of developing

a method and apparatus for digitizing multiple data streams whose clocks may vary

due to oscillator drift.

SUMMARY OF THE INVENTION

[0009]   The present invention solves these and other problems by providing a

method and apparatus for digitizing multiple data streams having different clocks in

which an error due to clock drift is spread across many clock cycles in extremely

small amounts.

[0010]   According to one aspect of the present invention, a method for

combining two data streams interpolates one or more samples between existing

samples of one of the two data streams and then adjusts a number of samples of said

one of the two data streams to maintain balance in a downstream synchronizing

buffer. This occurs prior to combining the two data streams. The adjusting may be performed by adding or decimating samples from the interpolated samples.

[0011]   According to another aspect of the present invention, an apparatus for combining two data streams includes two buffers, an interpolating filter, a multiplexer, a buffer controller and an adder. A first of the two buffers receives a first data stream of the two data streams and has its input clocked in by a first sample clock associated with the first data stream and has its output clocked out by the first sample clock. The interpolating filter receives the second data stream and outputs a decimated oversampled version of the second data stream. The multiplexer has its first input coupled to an output of the interpolating filter, receives the second data stream at its second input and outputs a modified data stream. A second of the two buffers receives the modified data stream, and has its input clocked in by a second sample clock associated with the second data stream, has its output clocked out by the first sample clock. The second buffer includes a level monitor output. The buffer controller has an input coupled to the level monitor output of the second buffer, has a first output controlling an output of the multiplexer, has a second output controlling the output of the interpolating polyphase filter and has a third output controlling the output of the second buffer. The adder then combines the outputs of the first and second buffer.

[0012]   According to yet another aspect of the present invention, a method for combining two asynchronous data streams having clocks offset in frequency four steps. First, a first data stream is clocked into and out of a first buffer using a first clock associated with the first data stream. Second, a second data stream is clocked

into a second buffer using a second clock associated with the second data stream and clocking the second data stream out of the second buffer using the first clock. Third, samples are interpolated into and decimated from samples of the second data stream prior to clocking the second data stream into the second buffer based on an overflow or underflow of the second buffer. Finally, the outputs of the first and second buffers are combined. In addition to the above four steps, a sample from the second buffer may be dropped when a buffer level of the second buffer increases by one sample by, e.g., disabling writing into the second buffer when a buffer level of the second buffer increases by one sample. Moreover, a sample may be added to the second buffer when a buffer level of the second buffer decreases by one sample by, e.g., disabling reading out of the second buffer when a buffer level of the second buffer increases by one sample.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013]   FIG 1 depicts a block diagram of an exemplary embodiment of an apparatus according to one aspect of the present invention.

[0014]   FIG 2 depicts a flow chart of an exemplary embodiment of a method according to another aspect of the present invention.

DETAILED DESCRIPTION

[0015]   It is worthy to note that any reference herein to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one

embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

[0016]   According to one aspect of the present invention, a method is disclosed that allows two data streams to be summed in a manner that results in acceptable degradation from the two different sample rates being slightly offset due to, e.g., normal clock drift. In one sense, the method of the present invention spreads the error due to difference clock drifts across many clock cycles by applying an extremely small amount of error across those many clock cycles rather than one large error over a single clock cycle.

[0017]   According to one embodiment of the present invention, the method interpolates (i.e., adds more samples between existing samples) one of the two data streams through the use of a modified polyphase, interpolating filter. Samples are then added or dropped as needed to keep a downstream synchronizing buffer (e.g., a FIFO) balanced. As the output of the downstream buffer is clocked by the non-interpolated data stream, balance is achieved by adding or removing samples to or from the interpolated data stream. As a result, the two streams are synchronized for subsequent summation. The interpolated data stream is then decimated (i.e., samples are removed) to the original sample rate and then fed into the synchronizing FIFO. Data is then read from the two FIFOs, summed together, and sent to the fiber optic circuitry for transmission.

[0018]   FIG 1 illustrates a block diagram of an exemplary embodiment 10 of an

apparatus for performing the data stream synchronization. Sampled data input 1 at n points per cycle is input to FIFO1 3 along with sample clock 1, which is used to clock the data into FIFO1 3 and to clock the data out of FIFO1 3. The output of FIFO1 3 is fed to one input of adder 1, the output of which adder 1 is the combined data that is output to the laser transmitter that is part of the return data path. The output of FIFO1 3 is sampled data at approximately n points per cycle.

[0019]   Sampled data input 2 is fed into one input of a multiplexer 5 and also into a modified polyphase interpolator $\times$ m filter 11. The two sample clocks and data streams are asynchronous and up to 10 parts per million offset in frequency.

[0020]   The output of the modified polyphase interpolator $\times$ m filter 11 is fed to the second input of the multiplexer 5. The output of the modified polyphase interpolator $\times$ m filter 11 is decimated sampled data at approximately n points per cycle. The output of multiplexer 5 is fed into FIFO2 2, whose input is clocked in with sample clock 2. The output of FIFO2 2 is clocked out into a second input of adder 1 by sample clock 1, which is also used to clock the out the output of FIFO1 3. Clocking outputs of both FIFO1 3 and FIFO2 2 with clock 1 will synchronize the two data streams but FIFO2 2 may underflow or overflow if sample clock 2 is not <u>exactly</u> the same as clock 1.

[0021]   The FIFO monitoring and control circuit 4 functions by watching the number of samples in FIFO2 2. As long as the number of samples in FIFO2 2 is between the established upper and lower thresholds, then the monitor circuit 4 is in its normal state. If the number of samples falls below the lower threshold, then the monitor circuit 4 detects an underflow and enables the polyphase filter 11 to add

samples until the number of samples in FIFO2 2 is above the lower threshold again.

If the number of samples in FIFO2 2 rises above the upper threshold, then the

monitor circuit 4 detects an overflow and enables the polyphase filter 11 to drop

samples until the number of samples in FIFO2 2 is again below the upper threshold

[0022]   When the FIFO monitoring and control circuit 4 does not detect an

overflow or underflow condition, no correction is needed. Therefore, it sets the

input multiplexer 5 to allow the data input 2 to flow directly into FIFO2 2.

[0023]   Turning to FIG 3, shown therein is an exemplary embodiment of a

correction process 30 according to one aspect of the present invention. The process

30 begins by detecting a number of samples in the buffer (step 31). When an

overflow is detected in step 32, the monitor circuit 4 enables the polyphase filter 11

to drop a sample. On the first cycle, the input multiplexer (IMUX) 5 is set to input

2 (step 32). The input multiplexer 5 is kept in this state for rest of the correction

process. Also in the first cycle, the filter multiplexer (FMUX) 6 is set to input 1

(step 32). This allows the value from the Phase1 filter (PF) 7 to be loaded into

FIFO2 2. On the second cycle, the filter multiplexer 6 is set to input 2. This allows

the value from the Phase 2 filter 8 to be loaded into FIFO2 2. On each successive

cycle, the next higher input of the filter multiplexer 6 is selected so that values from

each of the phase filters is loaded into FIFO2 2 (step 33). On cycle 256, the filter

multiplexer 6 is set to input 256, and the value from Phase 256 filter 9 is loaded into

FIFO2 2 (step 34). On cycle 257, the input multiplexer is set to input 1 and the

filter multiplexer 6 is set to input 1 (step 35). Also during cycle 257, the write

enable of the FIFO2 2 is disabled so that no value can be written into FIFO2 2 (step

35). At this point, a sample has been dropped and the correction process is complete and the process returns to step 31. On cycle 258, the write enable of FIFO2 2 is enabled and new samples can be loaded. The monitoring circuit 4 will either return to its normal state or, if an overflow condition still exists, then the monitor circuit 4 will again enable the correction process to drop another sample.

[0024]   When an underflow is detected (step 32), the monitor circuit 4 enables the polyphase filter 11 to add a sample. On the first cycle, the input multiplexer 5 is set to input 2 (step 36). It will stay in this position for the rest of the correction process. Also during the first cycle, the filter multiplexer 6 is set to input 256 (step 36). During this first cycle, the regular sample input and the value from the Phase 256 filter 9 are both loaded into the FIFO2 2 (step 36). This requires a special implementation of FIFO2 2 to allow this ability. On the second cycle, the filter multiplexer 6 is set to input 255 so that the value from the Phase 255 filter is loaded into FIFO2 2 (step 37). On each successive cycle, the next lower input of the filter multiplexer 6 is selected(i.e., the output of the next lower phase filter is selected as the input to the filter multiplexer 6) (step 37). In this way all of the Phase filters are cycled through in reverse order and values of each Phase filter is loaded into FIFO2 2 (step 37). On cycle 256, the filter multiplexer 6 is set to input 1 and the value from Phase 1 filter 7 is loaded into FIFO2 2 (step 37). On cycle 257, the input multiplexer 5 is reset to input 1 (step 38). This completes the correction process and the process returns to step 31. The monitoring circuit 4 will either return to its normal state or, if an underflow condition still exists, then the monitor circuit 4 will again enable the correction process to add another sample.

[0025]   Interpolating filter 11 includes m phases 7-9, the outputs of which are fed into a multiplexer 6, which provides the output of the interpolating filter 11 to multiplexer 5. An possible implementation uses 256 phases. In essence, interpolating filter adds samples and then decimates samples based on the overflow and underflow of FIFO2 2 to add small amounts of delay over many clock cycles to account for the relative clock drift between sample clock 1 and sample clock 2. The output of interpolating filter 11 is decimated sampled data at approximately n points per cycle.

[0026]   Interpolation and decimation are well-established techniques for rate matching one data stream to another, hence will not be explained in detail herein. In the text book case, a rate change also implies that the new data stream would be clocked at the higher rate (for the interpolation case), a sample dropped and then the interpolated data stream would be decimated back down to the 100MHz sample rate.

[0027]   The techniques described herein do not increase the clock rate because the maximum clock speed of an FPGA is about 120 MHz. In this application, by starting with a 100 MHz sample rate, interpolating by 256 would result in a sample clock rate of 25.6 GHz, an unattainable value. The polyphase filter structure allows us to have 256 new phases of the original 100 MHz data running in parallel.

[0028]   This presents a new problem, however, as building a 256-polyphase filter would require many FPGA devices (approximately 100). To conserve design size, without loss of performance, an exemplary embodiment of the present invention uses an FIR interpolating filter whose coefficients are stored in RAM that

is part of the FPGA device. For interpolation by 256, the embodiment stores 256

sets of coefficients in memory. For an 8-tap FIR interpolating filter, each set

would have 8 coefficients. This requires about 30% of the available RAM on a

medium size FPGA.

[0029]   For this embodiment, the modified polyphase filter 11 is changed.

Instead of 256 Phase filters and a 256-to-1 multiplexer, the polyphase filter 11

contains one phase filter that can load its coefficients from a RAM. Instead of

changing the input of the filter multiplexer 6 to switch from one interpolating phase

to the next, the embodiment switches a new set of coefficients into the phase filter.

[0030]   To keep the system running at real-time speeds without the use of a

25.6 GHz clock, 255 phase delays must be introduced, followed by dropping of a

sample, and then decimating. This technique essentially introduces 256 extremely

small phase errors that are spread over many clock cycles instead of one big phase

error in one clock cycle. In this manner, the phase error introduced is so small that

it negligibly impacts performance. For example, if the RF signal being sampled

with a 100 MHz A/D is centered at 10 MHz, then there are 10 samples per period.

Dropping one sample out of 2560 samples per period is equal to a 0.14 degree

(360°/2560 samples) of error – an insignificant value of phase step. Furthermore,

this error is spread over a time period associated with the clock offsets. In this way,

communication performance from the nodes is not impacted.

[0031]   FIG 2 depicts an exemplary embodiment 20 of a method according to

another aspect of the present invention. To combine two asynchronous data

streams in which the relative clocks are offset in frequency due to normal clock

drift by an amount such as 10 parts per million, the two data streams must be synchronized.

[0032]   To do so, in step 21 the first data stream is clocked into and out of a first buffer using a first clock associated with the first data stream. In step 22, the second data stream is clocked into a second buffer using a second clock associated with the second data stream. The second data stream is then clocked out of the second buffer using the first clock. This synchronizes the two data streams. If little or no offset in frequency existed, this would be sufficient. However, due to the offset, some adjustment is necessary.

[0033]   In step 23, samples of the second data stream are interpolated and then decimated prior to clocking the second data stream into the second buffer based on an overflow or underflow of the second buffer, which is downstream from the interpolation and decimation process.

[0034]   In step 24, samples are dropped from the second buffer when a buffer level of the second buffer increases by one sample, e.g., by disabling writing into the second buffer when a buffer level of the second buffer increases by one sample.

[0035]   In step 25, samples are dropped from the second buffer when the buffer level of the second buffer decreases by one sample, e.g., by disabling reading out of the second buffer when a buffer level of the second buffer increases by one sample.

[0036]   Finally, in step 26 the outputs of the first and second buffers are combined to provide an input for the return data path.

[0037]   Thus, the present invention provides an approach for digitally summing

geographically diverse data streams, which approach is consistent with the nature of HFC architecture. Unique application of DSP algorithms, and modeling and analysis of performance impacts, show a clear solution to the problem of distributed clocks in an HFC plant. Furthermore, the approach applies to any DSP functionality that is desired to apply to data stream derived from asynchronous clocks.

[0038] Although various embodiments are specifically illustrated and described herein, it will be appreciated that modifications and variations of the invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention. For example, while several of the embodiments depict the use of specific data formats and protocols, any formats or protocols may suffice. Moreover, while some of the embodiments describe specific embodiments of computer, clients, servers, etc., other types may be employed by the invention described herein. Furthermore, these examples should not be interpreted to limit the modifications and variations of the invention covered by the claims but are merely illustrative of possible variations.